Programming: Visual Basic for Applications Kick-off

The small print

Prerequisites

Time in the workshop is precious – it is an opportunity for you to interact with the workshop leader and other participants through questions and discussions and to share your experiences and concerns. To make the most of this time we sometimes ask you to carry out learning activities ahead of the workshop so that everyone comes into the class with the same basic knowledge. We keep this prior learning to a minimum and often make use of online videos. Online videos provided through 'Molly' can be accessed by University members anytime, anywhere, through a browser or app.

Your course booking will tell you if any prior learning activity is required. If you don't have an environment where you can do this learning, you can come along to one of our 'qu**i**e**t**' sessions. These are scheduled every week in normal term-time, and are a quiet space where you can work through 'Molly' videos or other workshop resources.

If you turn up for a workshop without having done the prior learning, the workshop leader may suggest that you come back on another session.

Copyright

About the workshop designer

Graham Addis started his first technology role in 1978 and has gathered decades of practical experience in industry. He has always been passionate about passing on his knowledge and undertook his first formal teaching position as a Customer Training Specialist for Intel back in 1984. Since that time his career has combined extensive real world experience with teaching and mentoring. In 2017 he joined the academic world at the University of Oxford and currently specialises in teaching spreadsheets, databases and programming.

Revision history

Version	Date	Author	Comments
2.0	June 2020	Graham Addis	Convert to online format.
1.0	November 2019	Graham Addis	Created

About this workshop

This workshop will take you through the basics of VBA Macros, using Excel as an example environment. We will provide you with an introduction with which you can use readily available learning resources to build on and expand the knowledge gained in this course.

We will include pointers to other workshops and further resources that will help you go on later to master VBA Macros.

What you will learn

We will start with the structure of the Visual Basic Component model used within all the Microsoft Applications.

We will investigate reference and manipulate cells and other objects within the VBA environment.

Although Visual Basic is relatively straightforward, applying it to the objects within the Microsoft Application suite can be confusing. By providing guided examples we will familiarise you with the basics of editing, running and modifying VBA Macros.

What you need to know

The ideas and techniques covered in this workshop will apply to a range of applications. We will demonstrate using *Excel for Windows*. However, the concepts will be the same, whatever Visual Basic enabled application you decide to use.

I will assume that you are familiar using Excel You will need to be able to:

- open and navigate around a workbook using the mouse and scrollbars, save a workbook
- add data to cells, and select and amend such data
- navigate the commands and menus, using Help as necessary

If you need to review these activities, Molly is a great place to get guidance. There are a large number of VBA related online-videos available for University members.

The resources you need

Sample documents that you can use to experiment with will be made available, but you may like to bring along your own.

Unless you have been told otherwise, in classroom workshops there will be a computer available for you to use with *Excel for Windows* installed.

You can use your own computer with your preferred app installed if you want to – just bear in mind that I am not an expert in every app (although I am sure that between us we will be able to sort out most problems!).

Study Videos

During the workshop, I will point you to a variety of resources that will help you in achieving these objectives.

Videos to support these topics are available through Molly – the IT Learning Centre's collection of online courses and resources, including the University's subscription to LinkedIn Learning. Read about Molly here: https://skills.it.ox.ac.uk/molly.

Learning Objectives

This workshop has the following learning objectives:

Learning Objective One - Accessing the Visual Basic development environment

Learning Objective Two - Creating a macro

Learning Objective Three - Saving a workbook with Macros

Learning Objective Four - Examine a working macro

Learning Objective Five - Extending a working macro

Learning Objective One - Accessing the Visual Basic development environment

There are a few ways to access the Visual Basic development environment within Excel. The simplest is by navigating to **View->Macros->View Macros**, then select ‘**Edit**’

An alternative is to enable the ‘**Developer**’ tab using ‘**File->Options->Customize Ribbon**’ option. It is then possible to select **Developer->Visual Basic**.

Learning Objective Two - Creating a macro

Macros exist within ‘**Modules**’.

Using the workbook **ChartData.xlsx** open the Visual Basic development environment and add a new module. **Insert->Module**.

Add a subroutine by inserting the following text into the new Module:

```
Sub test()
    ActiveSheet.Shapes.AddChart
End Sub
```

Run the macro **Run->Run Sub/UserForm** and see the results in the worksheet **ChartData**.

Learning Objective Three - Saving a workbook with Macros

Save the **ChartData.xlsx** workbook using **File->Save**, note the warning dialogue and adjust the file type on save.

Learning Objective Four - Examine a working macro

Open the excel enabled workbook **TestScoresMacro.xlsm** in the worksheet **TestScores**.

Examine and run the ‘**Charter**’ VBA subroutine.

Learning Objective Five - Extending a working macro

Open the ‘excel enabled’ workbook **TestScoresMacro.xlsm** in the worksheet **TestScores**.

Extend the ‘**Charter**’ VBA subroutine to include scores for **Mathematics** in the chart.

Further information

Getting extra help

Course Clinics

The IT Learning Centre offers bookable clinics where you can get pre- or post-course advice. Contact us using courses@it.ox.ac.uk.

Study Videos from Molly

Molly is our collection of self-service courses and resources. This includes providing LinkedIn Learning video-based courses free to all members of the University. Visit skills.it.ox.ac.uk/molly and sign in with your Single Sign-On (SSO) credentials.

Some courses recommend pre- and/or post-course activities to support your learning. You can watch these online videos anywhere, anytime, and even download them onto a tablet or smartphone for off-line viewing.

If you need a quiet place to work through learning activities away from distractions, the IT Learning Centre offers 'quiet' sessions where you can book a place. These are scheduled frequently during normal term times.

About the IT Learning Portfolio online

Many of the resources used in the IT Learning Centre courses and workshops are made available as Open Educational Resources (OER) via our Portfolio website at skills.it.ox.ac.uk/it-learning-portfolio.

About the IT Learning Centre

The IT Learning Centre delivers over 100 IT-related teacher-led courses, which are provided in our teaching rooms and online, and we give you access to thousands of on-line self-service courses through Molly (powered by LinkedIn Learning).

Our team of teachers have backgrounds in academia, research, business and education and are supported by other experts from around the University and beyond.

Our courses are open to all members of the University at a small charge. Where resources allow, we can deliver closed courses to departments and colleges, which can be more cost-effective than signing up individually. We can also customize courses to suit your needs.

Our fully equipped suite of seven teaching and training rooms are usually available for hire for your own events and courses.

For more information, contact us at courses@it.ox.ac.uk

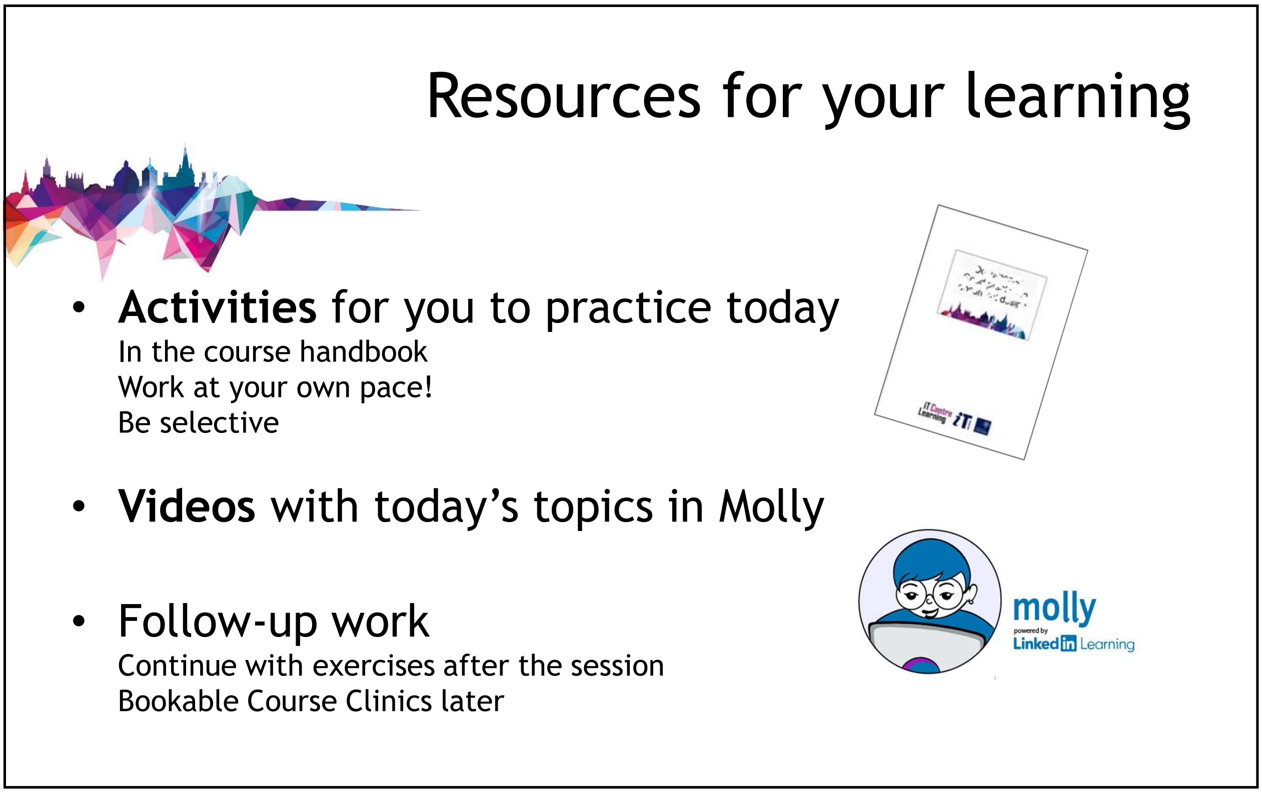
About IT Customer Services

The IT Learning Centre is part of the Customer Services Group. The group provides the main user support services for the department, assisting all staff and students within the University as well as retired staff and other users of University IT services. It supports all the services offered by IT Services plus general IT support queries from any user, working in collaboration with local IT support units.

The Customer Services Group also offers a data back-up service; an online shop; and a PC maintenance scheme. Customer Services is further responsible for desktop computing services – for staff and in public/shared areas – throughout UAS and the Bodleian Libraries.

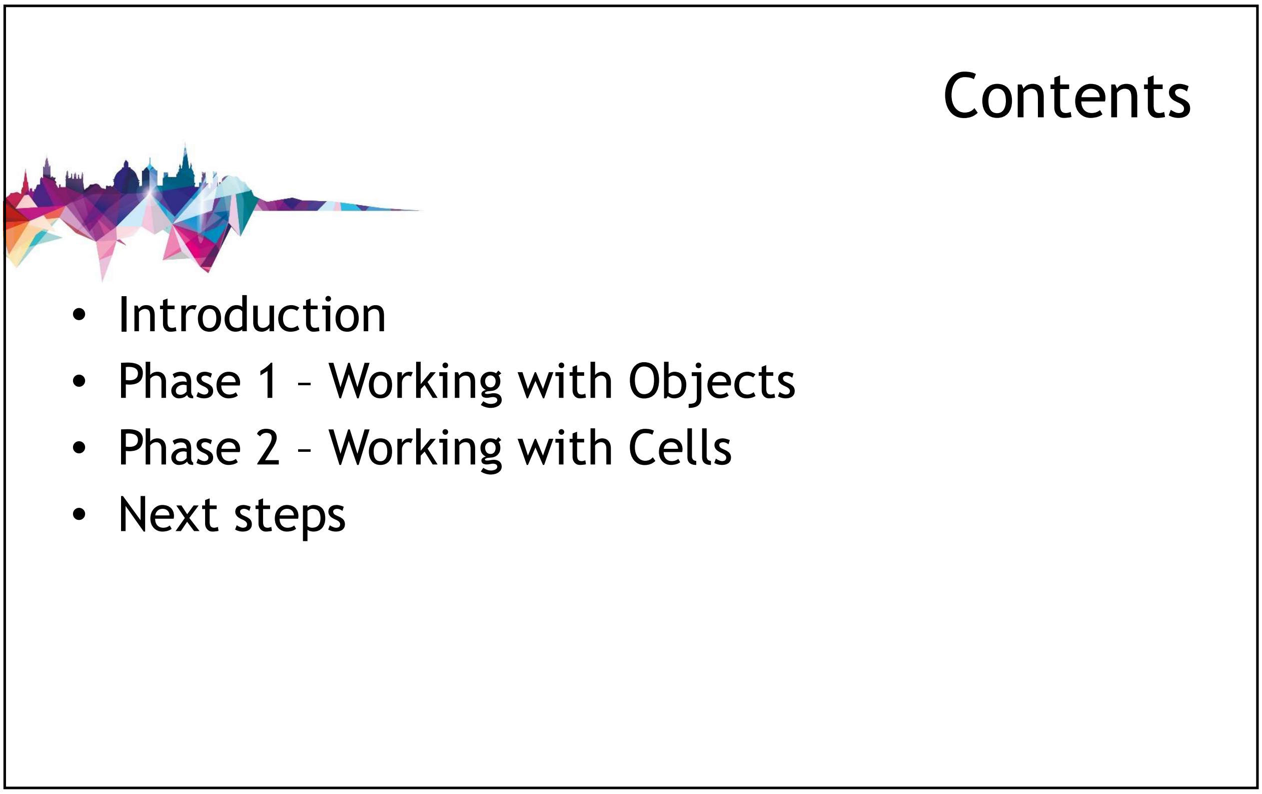
VBA Kick-Off Session

Graham Addis
graham.addis@it.ox.ac.uk



Resources for your learning

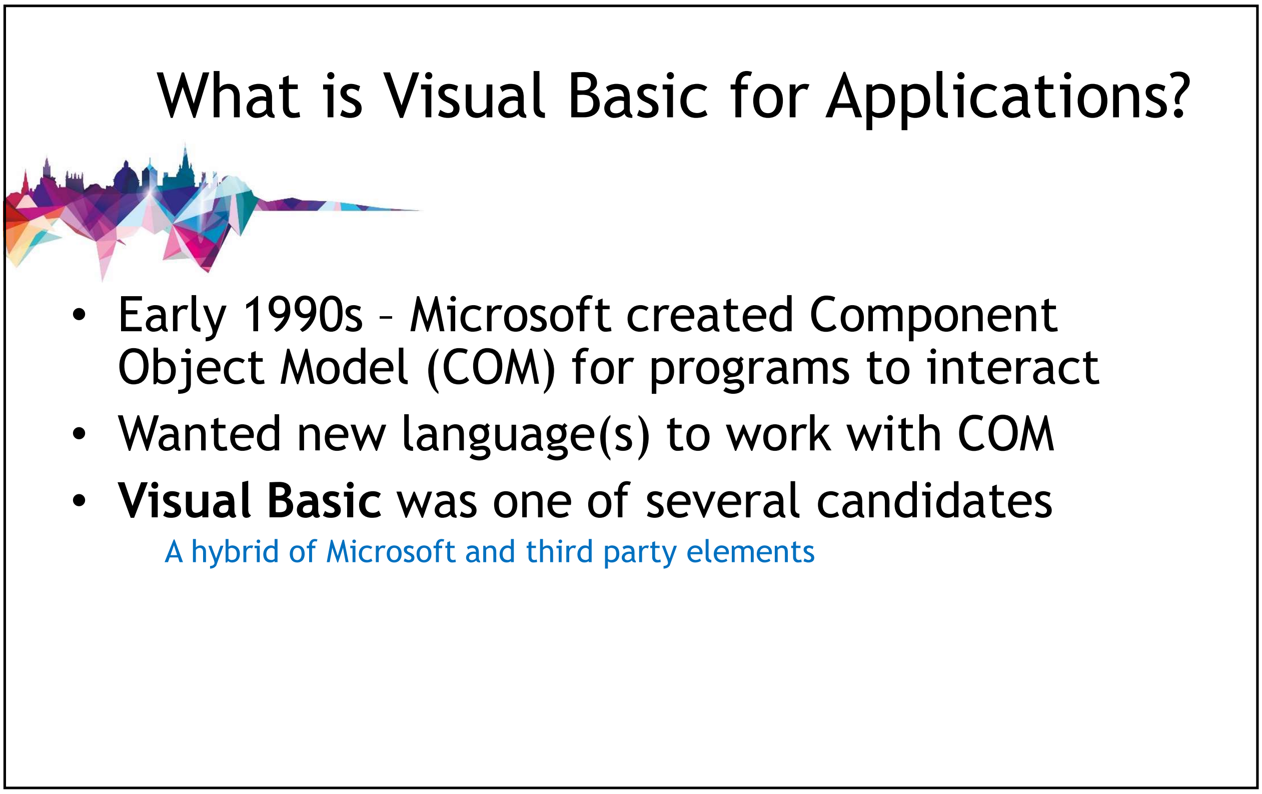
- **Activities** for you to practice today
  In the course handbook
  Work at your own pace!
  Be selective
- **Videos** with today's topics in Molly
- Follow-up work
  Continue with exercises after the session
  Bookable Course Clinics later



Contents

- Introduction
- Phase 1 – Working with Objects
- Phase 2 – Working with Cells
- Next steps

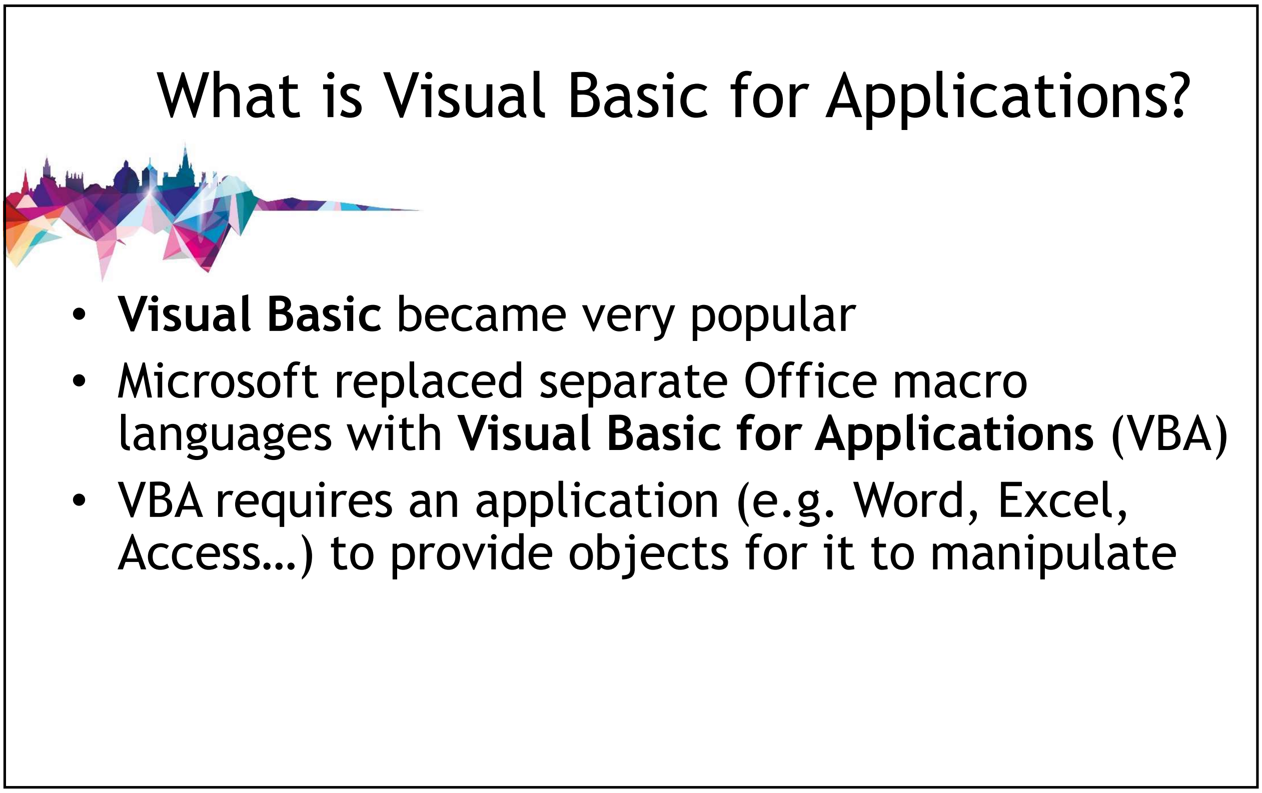
Introduction



What is Visual Basic for Applications?

- Early 1990s – Microsoft created Component Object Model (COM) for programs to interact
- Wanted new language(s) to work with COM
- **Visual Basic** was one of several candidates

A hybrid of Microsoft and third party elements



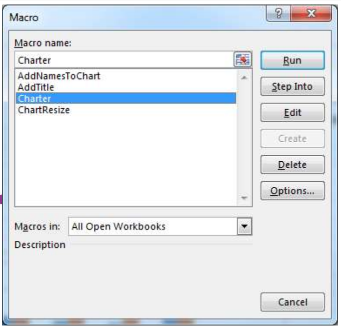
What is Visual Basic for Applications?

- **Visual Basic** became very popular
- Microsoft replaced separate Office macro languages with **Visual Basic for Applications** (VBA)
- VBA requires an application (e.g. Word, Excel, Access…) to provide objects for it to manipulate

Introduction

- Exploring a small example program
- Based on principles in Programming Concepts course
- Discussion encouraged (feedback very desirable)

Demo of Program

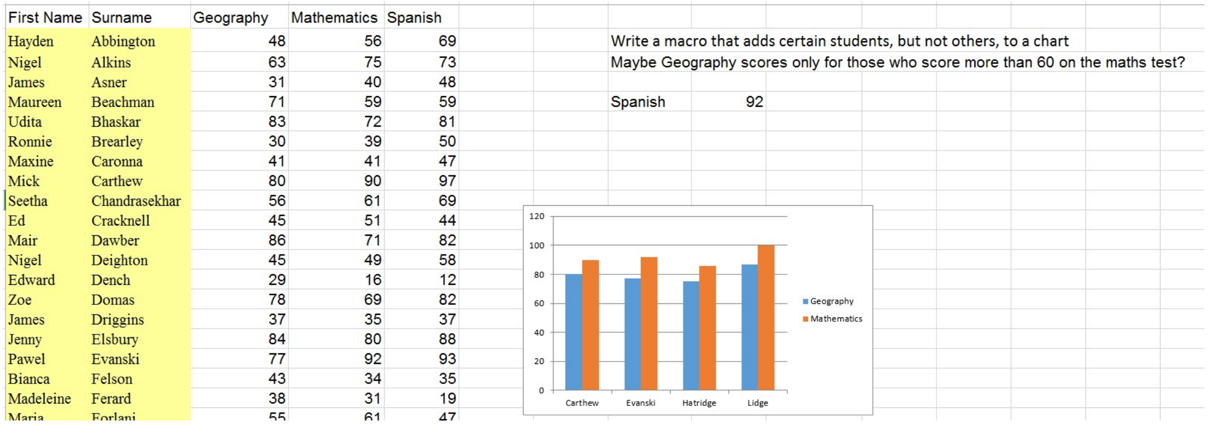


Macro name: Charter

AddNamesToChart
AddTitle
Charter
ChartResize

Macros in: All Open Workbooks

Description



First Name	Surname	Geography	Mathematics	Spanish
Hayden	Abbington	48	56	69
Nigel	Alkins	63	75	73
James	Asner	31	40	48
Maureen	Beachman	71	59	59
Udita	Bhaskar	83	72	81
Ronnie	Brearley	30	39	50
Maxine	Caronna	41	41	47
Mick	Carthew	80	90	97
Seetha	Chandrasekhar	56	61	69
Ed	Cracknell	45	51	44
Mair	Dawber	86	71	82
Nigel	Deighton	45	49	58
Edward	Dench	29	16	12
Zoe	Domas	78	69	82
James	Driggins	37	35	37
Jenny	Elsbury	84	80	88
Pawel	Evanski	77	92	93
Bianca	Felson	43	34	35
Madeleine	Ferard	38	31	19
Maria	Forlani	55	61	47

Write a macro that adds certain students, but not others, to a chart

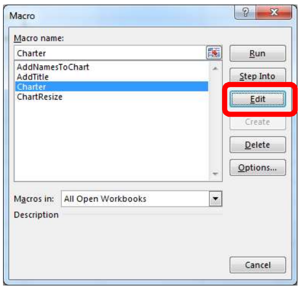
Maybe Geography scores only for those who score more than 60 on the maths test?

Spanish	92

Phase 1 – Working with Objects

Find the code window

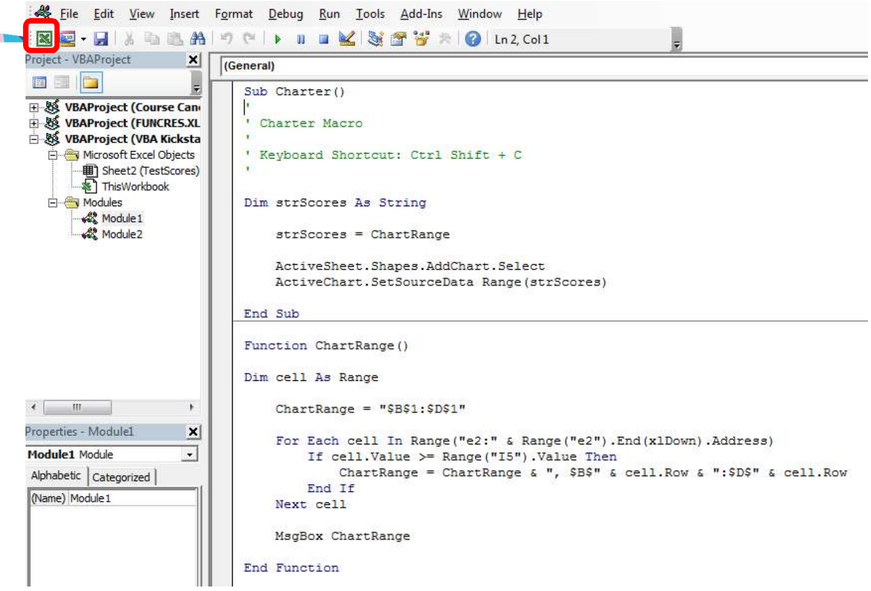
- Alt+F11



Macro name: Charter

AddNamesToChart
AddTitle
Charter
ChartResize

Macros in: All Open Workbooks



VBAProject (Course Can
VBAProject (FUNCRES.XL
VBAProject (VBA Kicksta
Microsoft Excel Objects
Sheet2 (TestScores)
ThisWorkbook
Modules
Module1
Module2

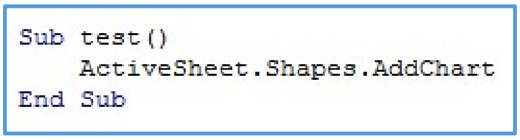
Module1 Module

(Name) Module1

(General)

```
Sub Charter()
'
' Charter Macro
'
' Keyboard Shortcut: Ctrl Shift + C
'

Dim strScores As String

    strScores = ChartRange

    ActiveSheet.Shapes.AddChart.Select
    ActiveChart.SetSourceData Range(strScores)

End Sub

Function ChartRange()

Dim cell As Range

    ChartRange = "$B$1:$D$1"

    For Each cell In Range("e2:" & Range("e2").End(xlDown).Address)
        If cell.Value >= Range("I5").Value Then
            ChartRange = ChartRange & ", $B$" & cell.Row & ":$D$" & cell.Row
        End If
    Next cell

    MsgBox ChartRange

End Function
```

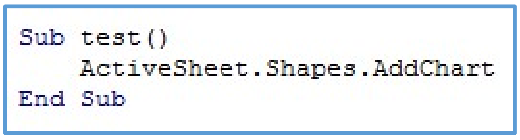
Set up a test Sub…



```
Sub test()
    ActiveSheet.Shapes.AddChart
End Sub
```

- Type this in below the existing code
- Use Play button or F5 to test
- *What does it do?*

Using the test Sub…



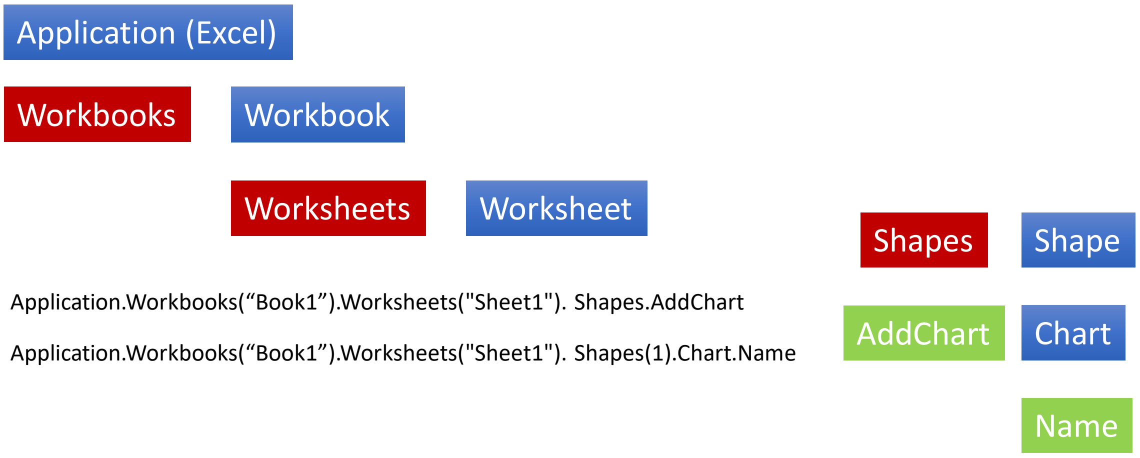
```
Sub test()
    ActiveSheet.Shapes.AddChart
End Sub
```

- *How many objects can you see mentioned here?*
- *What action is taking place? What general name do we give to it?*
- *What does it mean for an object to be “Active”?*

Practical Session 1

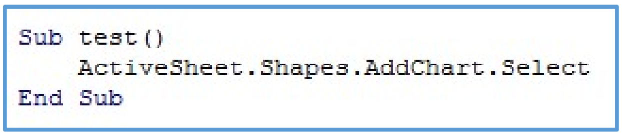
Learning Objective	Workbook	Worksheet
One	**ChartData.xlsx**	**ChartData**
Two	**ChartData.xlsx**	**ChartData**
Three	**ChartData.xlsx**	**ChartData**

Excel Object Model



```
Application.Workbooks(“Book1”).Worksheets("Sheet1"). Shapes.AddChart
Application.Workbooks(“Book1”).Worksheets("Sheet1"). Shapes(1).Chart.Name
```

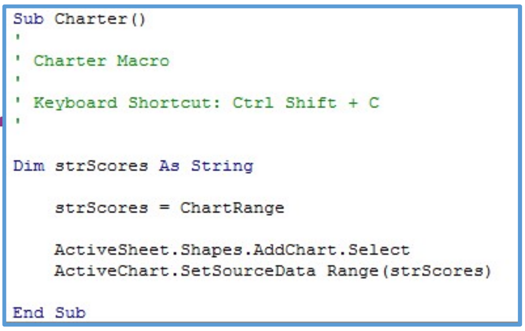
Using the test Sub…



```
Sub test()
    ActiveSheet.Shapes.AddChart.Select
End Sub
```

- *In code, do objects have to be selected for us to do something with them?*
- *Why might we select an object?*
- *Is Selecting the same as Activating?*

Charter

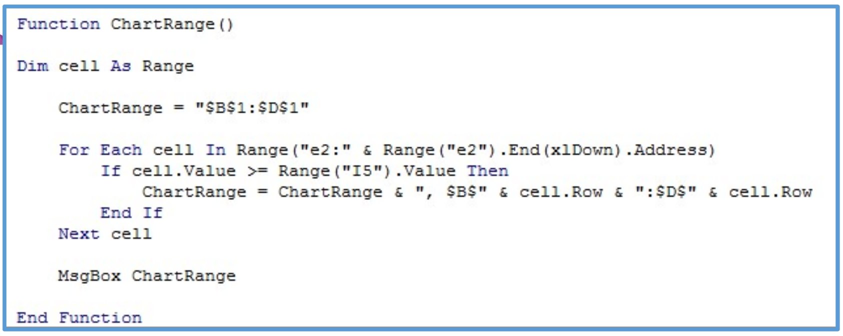


```
Sub Charter()
'
' Charter Macro
'
' Keyboard Shortcut: Ctrl Shift + C
'
Dim strScores As String

    strScores = ChartRange

    ActiveSheet.Shapes.AddChart.Select
    ActiveChart.SetSourceData Range(strScores)

End Sub
```

- *What is strScores? What are the two pieces to its name?*
- *What is ChartRange*
- *What are the green single quotes doing? Why might you use them?*

Phase 2 – Working with Cells

ChartRange



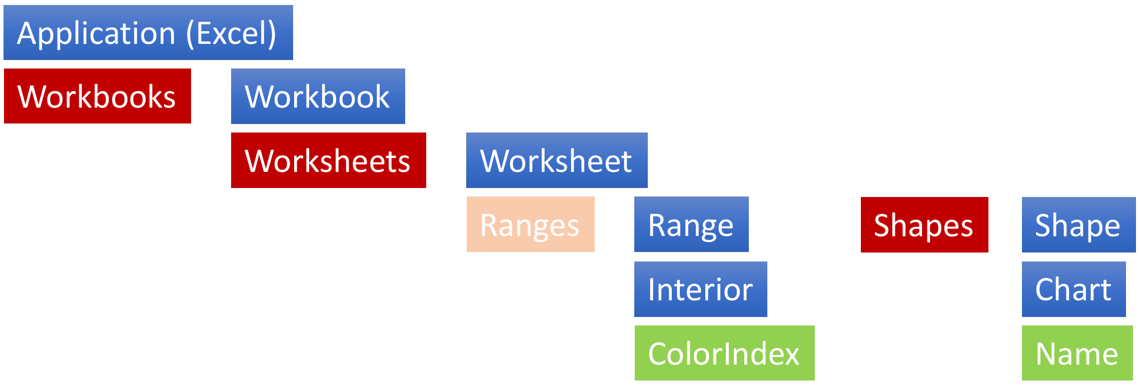
```
Function ChartRange()

Dim cell As Range

    ChartRange = "$B$1:$D$1"

    For Each cell In Range("e2:" & Range("e2").End(xlDown).Address)
        If cell.Value >= Range("I5").Value Then
            ChartRange = ChartRange & ", $B$" & cell.Row & ":$D$" & cell.Row
        End If
    Next cell

    MsgBox ChartRange

End Function
```

- ChartRange is a Function that returns a value
- *How does that make it different to Charter? (Look at View, Macros…)*

Working with Cells

- *How do we refer to cell J14 in code?*
- *What happens when you put a dot after that?*
- *What item(s) there could we use to write someone’s name into a cell?*
- *How would we complete that line?*
- *How could we do the same for a BLOCK of cells?*

Excel Object Model



Application (Excel)

Workbooks Workbook

Worksheets Worksheet

Ranges Range Shapes Shape

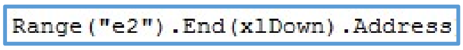
Interior Chart

ColorIndex Name

Application.Workbooks(“Book1”).Worksheets("Sheet1").Range("A1").Interior.ColorIndex = 8

Application.Workbooks(“Book1”).Worksheets("Sheet1"). Shapes(1).Chart.Name

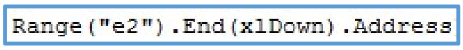
ChartRange



```
Range("e2").End(xlDown).Address
```

- *What sort of thing is End?*
- *What sort of thing is xlDown? How does it behave in the VBA editor?*

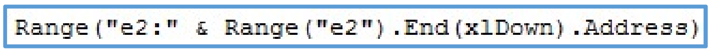
ChartRange



```
Range("e2").End(xlDown).Address
```

- Put MsgBox at the start of the line
- *What is MsgBox doing?*
- *Why is it useful here?*
- *Is it essential? If not what else could we use?*

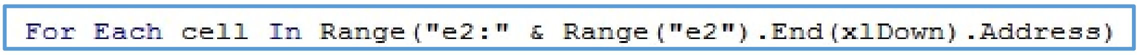
ChartRange



```
Range("e2:" & Range("e2").End(xlDown).Address)
```

- *What is the ampersand (&) doing here?*
- Test this with MsgBox at the start of the line
- *Are the brackets required?*
- *When do we need quotation marks?*

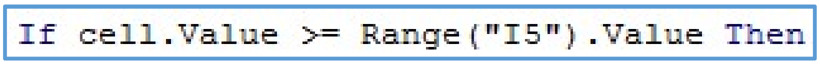
ChartRange



```
For Each cell In Range("e2:" & Range("e2").End(xlDown).Address)
```

- *Why is For Each useful here? What other kinds of loops are there?*
- *What sort of thing is "cell" here?*

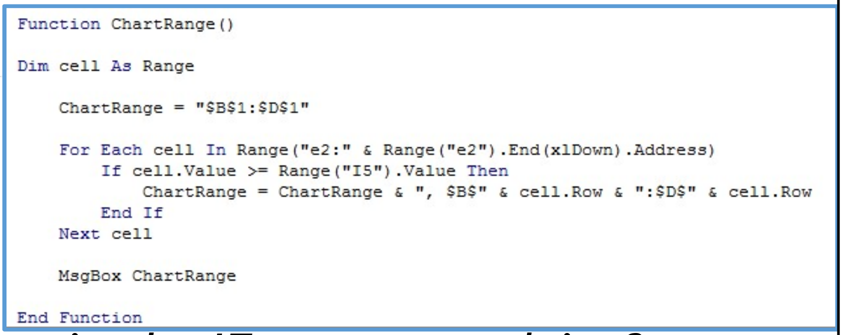
ChartRange



```
If cell.Value >= Range("I5").Value Then
```

- *What test are we doing here? Why do we need to do it?*
- *What sort of thing is Value?*
- *What is Then doing?*
- *What sort of thing is ">="*

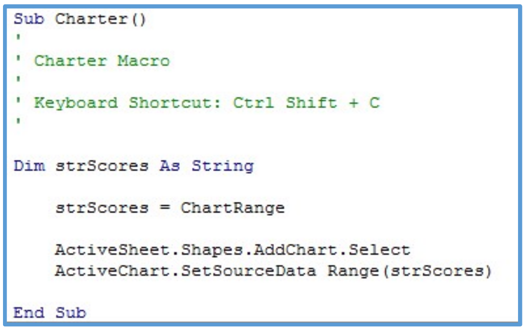
ChartRange



```
Function ChartRange()

Dim cell As Range

    ChartRange = "$B$1:$D$1"

    For Each cell In Range("e2:" & Range("e2").End(xlDown).Address)
        If cell.Value >= Range("I5").Value Then
            ChartRange = ChartRange & ", $B$" & cell.Row & ":$D$" & cell.Row
        End If
    Next cell

    MsgBox ChartRange

End Function
```

- *What is the line in the IF statement doing?*
- *What sort of thing is Row?*
- *What is ChartRange = "$B$1:$D$1" doing?*
- *How does the ChartRange value get back to Charter?*

Charter



```
Sub Charter()
'
' Charter Macro
'
' Keyboard Shortcut: Ctrl Shift + C
'

Dim strScores As String

    strScores = ChartRange

    ActiveSheet.Shapes.AddChart.Select
    ActiveChart.SetSourceData Range(strScores)

End Sub
```

- *What sort of thing is SetSourceData?*
- *How does the chart finally receive its data here?*

Practical Session 2

Learning Objective	Workbook	Worksheet
Four	**TestScoresMacro.xlsx**	**TestScores**
Five	**TestScoresMacro.xlsx**	**TestScores**

Next Steps

Next Steps

molly
powered by
LinkedIn LEARNING

COURSE POPULAR

Excel 2016: Macros in Depth

By: Dennis Taylor • Released Aug 16, 2016

COURSE POPULAR

Learning VBA in Excel

By: Curt Frye • Released Jan 22, 2019

80,209 viewers

Graham.Addis@it.ox.ac.uk